MULTIDIMENSIONAL SCHEMAS FOR ENGINEERING ASSET MANAGEMENT

Avin Mathew, Lin Ma
Cooperative Research Centre for Integrated Engineering Asset Management
School of Engineering Systems
Queensland University of Technology
Brisbane, Australia
a.mathew@qut.edu.au, l.ma@qut.edu.au

ABSTRACT
The amount of data that modern companies collect is ever increasing. With organisations that own a plethora of information systems, structured integration of data has become a pressing issue. Many companies have turned to data warehousing to bridge the gap of turning data into useful information and knowledge; however the data involved are typically from softer business domains rather than engineering-related domains. We investigate the area of engineering asset management data warehousing by examining the multidimensional modelling of asset management data. The multidimensional schemas are derived from the relational model in the MIMOSA OSA-EAI. The standards-based approach provides for platform neutrality, and as the OSA-EAI is a generic model, rules are given for company-specific implementations. From initial testing, query formation with multidimensional models is less complex than Third Normal Form models.

KEYWORDS
Asset management, data warehousing, multidimensional schema, MIMOSA

INTRODUCTION
Decision support systems often form the core IT requirements and infrastructure in a business because they give companies a way of turning knowledge into tangible results. The amount of data available to companies is often overwhelming, and collecting, maintaining and analysing the data requires significant organisational commitment. Many companies have turned to data warehousing to bridge the gap of turning data into knowledge. The data warehouse then forms the backbone for informational requirements to the decision support system. Serving as an information management solution that integrates information across domains, organisations, and applications, a data warehouse provides a conduit of accurate and timely information for analysis tools. In effect, it supports decision support systems. Physically, a data warehouse is a data repository devoted to analytical processing, as opposed to an online transaction processing (OLTP) database.

There have been several applications of data warehousing in engineering asset management firms. The term asset management is broad, and its activities are undertaken by a broad spectrum of firms. One common definition of asset management is “the systematic and coordinated activities and practices through which an organization optimally manages its physical assets and their associated performance, risks and expenditures over their lifecycles for the purpose of achieving its organizational strategic plan” [1]. A literature survey into asset management data warehousing showed that the asset lifecycle data was underutilised [2].

In this exploratory research into asset management data warehousing, we examined the suitability of multidimensional modelling of asset management data. Multidimensional modelling is the primary schema methodology used in data warehousing. These multidimensional schemas were derived from the Machinery Information Management Open Systems Alliance (MIMOSA) Open System Architecture for Enterprise Application Integration (OSA-EAI) version 3.0f. The MIMOSA OSA-EAI provides a standardised format for the arrangement of asset management data.

The advantages of multidimensional models typically stem from being integrated models that offer better performance over regular database models. There is also an abundance of Online Analytical Processing (OLAP) tools devoted to business intelligence that operate on multidimensional models. The primary reason of using the MIMOSA OSA-EAI is because it is a standardised model for engineering data, and the derived multidimensional models will not negate its main function of communications.

BACKGROUND THEORY

Data warehouse schema modelling

A schema is a description of the structure and rules of an object. In the data management sense of the word, it is the model that defines the data objects, their attributes, their relationships, and rules. There are several methodologies of arranging schema objects, with the most prevalent being the third normal form.

Third normal form – Third normal form (3NF) modelling is the classical approach to relational database design whereby data redundancy is minimised through normalisation. Because of normalisation, 3NF schemas typically have a larger number of tables compared to star and snowflake schemas.

Multidimensional schemas – There are two primary schema methodologies of representing multidimensional data, either as a star or snowflake schema. In a star schema, the data are stored in a central fact table, and surrounded by one or more denormalised dimension tables. It is named as such because of this central structure with radiating points. A snowflake schema is similar to a star schema, but allows for normalisation in dimensional tables to remove redundancy, and hence dimension tables can be associated with other dimension tables. Compared to a 3NF schema, multidimensional schemas are highly denormalised. Because of the decrease in complexity due to denormalisation, multidimensional schemas can be more intuitive to non-technical end users who are more familiar with logical entities rather than entities and relationships. They can also provide optimised performance for star queries, and there are a large number of business intelligence tools based around multidimensional schemas.

MIMOSA OSA-EAI

The absence of a standard for asset management data exchange was a driving factor in the formation of MIMOSA and the subsequent development of the OSA-EAI. The OSA-EAI provides open data exchange standards in several key asset management areas: asset register management; work management; diagnostic and prognostic assessment; vibration and sound data; oil, fluid and gas data; thermographic data; and reliability information. These seven areas are defined by a relational model named Common Relational Information Schema (CRIS). The CRIS defines asset management entities, their attributes and associated types, and also relationships between entities.

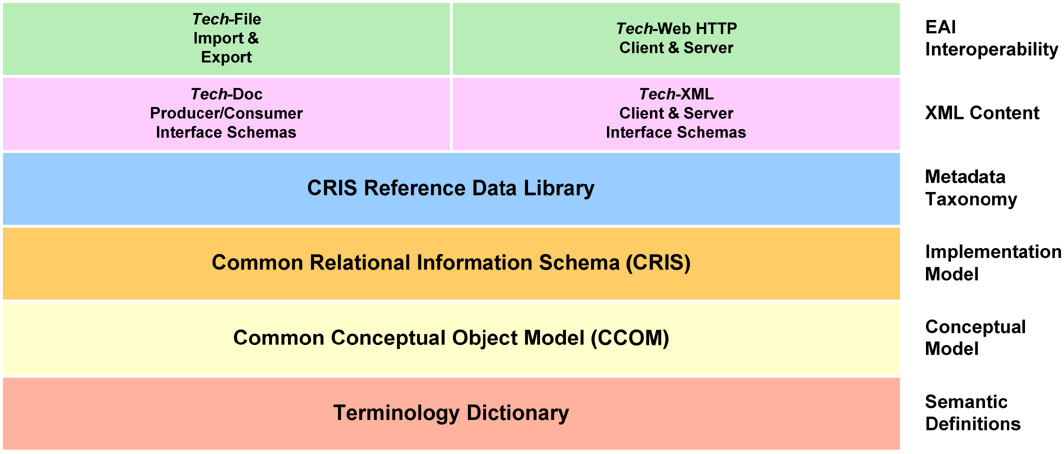

Figure 1 – MIMOSA OSA-EAI 3.0f layers

As seen in Figure 1, a reference data library sits on top of the CRIS. The library contains reference data compiled by MIMOSA which can be stored by the CRIS and are intended to facilitate communication between MIMOSA-compliant systems. The reference data primarily consist of 'type' information such as asset, segment, and event types. However, the largest component of the reference library is manufacturer details.

The OSA-EAI package contains SQL (Structured Query Language) scripts for creating a database based on the CRIS and inserting data from the reference library. A program does not need to implement the database component to be MIMOSA-compliant – only the XML schema must be implemented. However, a MIMOSA database implementation makes future development significantly easier in order to comply with the MIMOSA standards.

Entity relationship to multidimensional schema

There have been many methodologies [3-7] proposed by researchers to design multidimensional models from entity relationship (ER) diagrams. While the use of the term "ER diagram" is a misnomer as multidimensional models can also be represented as ER diagrams, for the purpose of this discussion, the term ER diagrams will be used interchangeably with 3NF. None of the methodologies are fully automatic, but require significant user interaction. In many cases, the methodologies focus on assisting the derivation of dimensions rather than facts. As with all automated or semi-automated data warehouse schema design techniques, these can be used as a starting platform for a designer who may be unfamiliar with the domain or underlying information systems.

MULTIDIMENSIONAL MODELLING

The methodology in this section is applied to the OSA-EAI CRIS with the approach being one that retains the structure of the original schema. Thus many entities are similar to their ones in the CRIS, but are collapsed into fewer tables due to denormalisation.

A multidimensional schema is not intended to be a complete replacement for a 3NF schema, but contains a subset of the data. There is a lot of data contained within the OSA-EAI that is unsuitable and unnecessary for data warehousing. Thus not all of the OSA-EAI is changed to a multidimensional form, instead, only those parts of the schema intended to be analysed become suitable candidates. Many of the semi-automated techniques that derive multidimensional schema from ER schema ignore this fact and attempt to include every entity in the derived schema, as the techniques cannot process business requirements. Hence entities such as Enterprise and Ordered List are not included in a multidimensional model as they contain little numerical factual information and are ineffective as a dimension.

As business requirements are a key point in schema development, the schemas presented are not the only construct that could be derived. Depending on requirement information, entities or attributes could be included or omitted, and terminology may be different. The differences are typically slight, and the discussion will highlight differences that may arise.

Methodology

As discussed earlier, there are no mature automated methodologies for deriving multidimensional schemas from an underlying information source. The methodology used in deriving the multidimensional schemas from the OSA-EAI CRIS is as follows:

1. *Identify fact tables by examining primary data tables*

There are several primary data tables that store frequently changing information. Reference tables such as `asset_type` and `segment` do not change often, while data tables such as `meas_event` and `work_order` have rows constantly added. These frequently changing tables are more likely to be fact tables, while reference tables are almost exclusively dimension tables.

2. *Identify potential dimensions by examining foreign keys of primary data tables*

Potential dimensions can be located through tables joined to primary data tables via a foreign key. These tables typically have attributes that can be used in dimensions for the primary data tables. In nearly all instances, these foreign key attributes have a finite domain of values, and this is a good indicator that is suitable as a dimension attribute.

When entity e_1 contains the attribute a, and a is a primary key of entity e_2, then a is designated as a foreign key. If $a \in A$ and $|A| \in \mathbb{Z}^+$, then e_1 is likely to be a dimension.

3. *Identify conformed dimensions by examining dimensions prevalent to multiple fact tables*

While dimension tables might consistently reoccur, the required attributes inside may change for different fact tables. Thus in identifying the reoccurring dimensions, the attribute set of the common dimension is the combined set of attributes.

When dimension d_1 joined to fact table f_1 and d_2 joined to fact table f_1 are logically equal, the common dimension formed is $d_c = d_1 \cup d_2$.

4. *Group dimension attributes outside of conformed dimensions into new dimension*

For those dimension attributes that cannot logically fit within a common dimension, these are inserted into a dimension specific to the fact table. While not quite a junk dimension, the grouping concept is similar.

5. *Identify fact attributes*

Several steps are used in identifying fact attributes. In step 2, attributes with a finite domain of values became dimension attributes; alternatively, numeric attributes that have an infinite domain of values are likely to be fact attributes. Pre-calculated attributes, such as differences between start and end times, and differences between scheduled and actual times are identified. As a lot of data is stored in linked binary/character/numeric tables, commonalities can be identified as fact attributes.

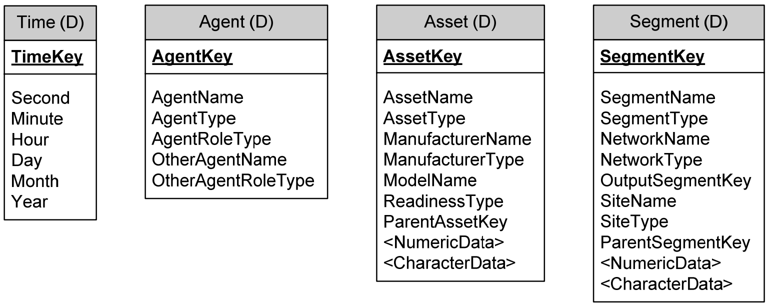

Figure 2 – Common conformed dimensions

Conformed dimensions

Conformed dimensions are the dimensions that have been standardised across multiple business units. While certain dimensions are more amenable to standardisation, others can be more subjective and vary according to business requirements. The advantage of standardising on dimensions is that data from different sources (either data warehouses/marts, or other facts) can be easily combined. This allows for the combination of a variety of data, leading to novel analysis approaches.

It was found that there are certain dimensions that are universal to nearly all areas asset management data. As it forms a founding dimension for almost all data warehouses, the Time dimension is an obvious observation. However, the dimensions of Asset, Location, and Agent were in most cases, just as ubiquitous.

Time Dimension – The lowest granularity for the Time dimension is that of a second. While time-based records formatted to the OSA-EAI specifications (which use ISO 8601) can represent fractions of a second, the fractional component is optional. Hence, the lowest required grain for a time record is a second. Other aggregations and derivatives attributes can be included in the dimension definition, such as Quarter or Week Number in Year attributes if deemed useful for decision support.

Agent Dimension – The definition of an agent is "an animate object (person, group, organization, or intelligent agent software) that makes various types of assessments" [8]. Agents are one of the simpler constructs within the OSA-EAI, only consisting of a type, a collection of roles, and roles with other agents.

Asset Dimension – The asset dimension combines asset, model, and manufacturer information attributes through denormalisation. It shows the first instance of how the pervasive entity-attribute-value (EAV) structures can be represented through a star schema. The OSA-EAI implements three common EAV constructs: those for numeric data, those for alpha-numeric data, and those for binary data. An asset is weakly typed entity whose real world attributes are mutable through the EAV structures. As a star schema moves towards a denormalised structure, the associated attributes in the EAV structure must be embedded within the asset dimension itself [9]. The schema designer is faced with the problem of determining common numeric and character data attributes to include in the asset dimension, hence why the attributes are marked with < > markers. As not all assets share the same attributes (e.g. a motor has a voltage rating but a pipeline does not), a compatible alternate solution is to subclass the asset entity into strongly typed entities. This procedure to increase flexibility snowflakes the schema, and as a result, the potential usability decreases.

Segment Dimension – The segment dimension contains the requisite segment name, type, and numeric and character data attributes in a similar vein to the asset dimension. Associated networks and sites are included, and depending on their implementation, they are not necessarily strict aggregations of segments.

Configuration data

Upon acquisition of an asset, details on the information of the asset are typically recorded in a register. These can include the purchase date and price, serial or model numbers, and technical characteristics of the asset. The relationship between assets and assemblies are also recorded. As in dimensional modelling, the characteristic data on assets are recorded in the dimension table as previously explained.

An asset's location is recorded in MIMOSA through the `asset_on_segment` table. It provides vital information on which segment each asset is located in an organisation. As with the original 3NF schema, the multidimensional version reports historical information on previous installations such that asset movement within a firm can be tracked over time. As with all time-related facts with designated start and end times, a pre-calculated duration fact is included to increase query speed. The schema also provides a method of analysing the scheduled and actual installation procedure duration. These two facts are not part of the original schema, but are often stored within work management records and are pertinent to assets that require non-trivial placement (i.e. installation). Subsequently, these fields require an ETL process for population.

Measurement

Measurements record the condition of an asset, which can consequently trigger a health assessment or register an alarm. Measurement events record information on the time a measurement was made at a specified measurement location, along with associated measured, transducer, and data source assets, data records, and confidence levels. Signal process streams represent the way data is processed. Data recorded through transducers are stored in separate tables aligned to their type which is defined by the corresponding meta-data. Hence time waveform data is stored distinctly to a single valued amplitude data.

The main fact, `Measurement Event`, is a denormalised multidimensional representation of the Measurement Event entity in the OSA-EAI schema. As with the asset and segment dimensions, the issue with numeric and character data is revisited. We also observe from Figure 3 that the number of keys significantly outweighs the number of non-null primary keys with this distinction being dictated by the original OSA-EAI model.

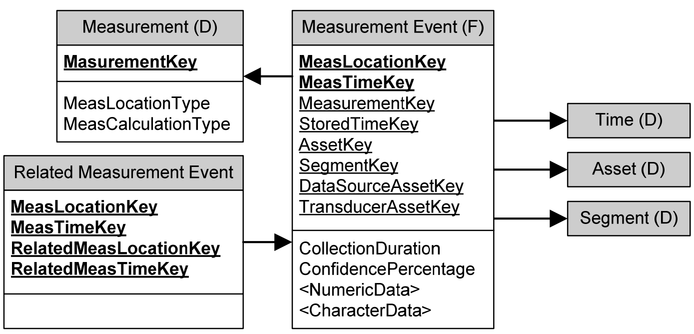

Figure 3 – Measurement Event star schema

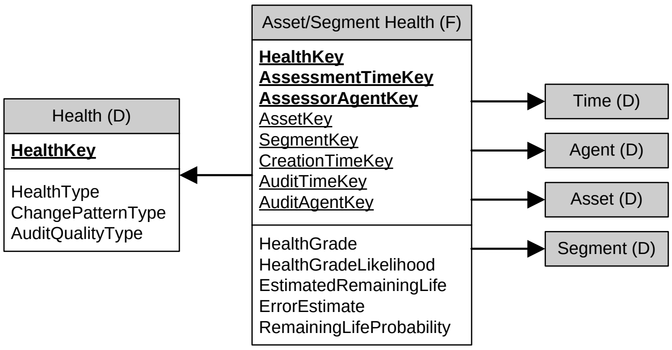

Figure 4 – Health data star schema

Measurement events can have associations with other measurement events, for example, collecting both vibration and RPM readings from a motor. This information is captured through the Measurement Event Association table in the OSA-EAI and through a non-dimension table, `Related Measurement Event`, for the star schema.

While contention exists over the semantics of the asset and segment values in a measurement event [10], asset installation information can be used to rebuild the association while using the schema. This allows users to use either the asset or segment dimensions directly without having to drill-across from an `Asset Installation` fact table.

Health and alarm data

Health and alarm data result from processing of measurement data collected from an asset or a segment. Health data in MIMOSA provide an indication on the condition of the entity in question, through codified health grades and change patterns, remaining life, and events that substantiate a health assessment. Alarm data record the measurement region that would trigger an alarm, and also any alarms registered.

Aggregations of health facts are more useful through more complex functions such as reliability functions rather than the standard statistical functions. For instance, health grades and remaining life in Figure 4 can be aggregated over assets or segments to provide a compound figure. For example, a pump system consists of a pump, motor, and shaft and a weighted average function can be used to estimate the health grade or remaining life of the entire pump.

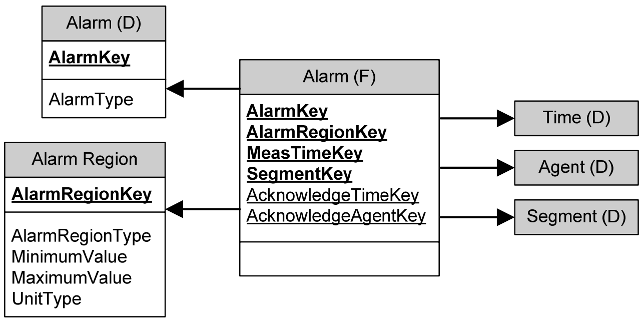

Figure 5 – Alarm data star schema

The `Alarm` fact table in Figure 5 is what is known as a factless fact table. Factless fact tables do not contain any numeric facts, but rather simply record an event through the association of different dimensions. While most queries on factless fact tables result in counts of alarms through different dimensions, queries can be conducted involving the time between alarm registration and alarm acknowledgement to determine the efficiency of business processes.

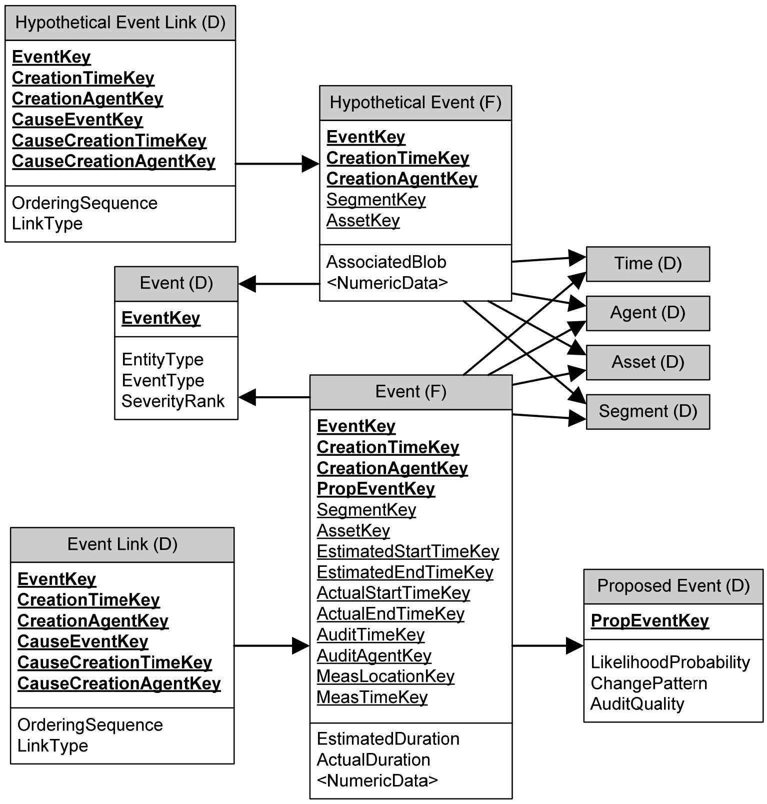

Figure 6 – Event star schema

Event

An event is a phenomenon that occurs at a single point in time and location. It can include anything that occurs in the physical world, including asset, human, and environmental activities. While measurement events and work order events also constitute as events, they are contained in special subclasses due to their regular occurrence in asset management operations.

Event tables in the OSA-EAI are divided into three discrete categories: those that could happen (hypothesised events), those that are scheduled to occur (proposed events), and those that have happened (actual events). These event types manifest in different locations (segment type, asset type, model, segment, and asset) and are attributed different combinations of characteristics linked through different tables.

The event star schema, shown in Figure 6, contains all of the data stored in the 45 event-related tables in the CRIS. The significant reduction in the number of entities comes through judicial selection of entities and attributes. By examining similarities in the structure of the type of events, Proposed and Actual events are combined into one fact table. This additionally allows for pre-calculated duration facts. In the `Event` dimension, the `entity type` attribute defines the occurrence location. The `link type` in the link dimensions defines the link characteristics.

The numeric data fields are taken from the `ev_num_data_type` table in the OSA-EAI, which stores 17 characteristics including MTBF and MTTR, safety and environmental impact ratings, and costs. As there are a comparatively small number of numeric data types, all utilised ones could be inserted as facts.

Work management

Work management is a core area of asset management as it forms the foundation of activities within an organisation. Nearly all firms that conduct asset management will implement work management systems at the very least, as quick gains in productivity can be harnessed by streamlining work management processes. The work management package in the OSA-EAI can be divided into three functions: work requests, work orders, and work order steps. Requests indicate a need for work, while orders detail the actual work performed. The work management schemas are fundamentally similar to the event schemas as described in the previous section, as work in itself can be considered as a type of event in the broad sense of the word. Both have the same time characteristics of scheduled and actual times, audit information, and asset and segment relations. In the OSA-EAI, they are distinguished by the intrinsic characteristics which are stored in the dimension table of the same name.

Work requests, work orders, and work order steps consist of similar attributes and are combined into one fact table. One fact table is used as the facts are very similar with the fundamental difference between requests and orders being the data contained in the ‘actual’ time attributes – as requests are issued before work is conducted, an ‘actual’ time value would be null. To distinguish requests, orders and order steps, an attribute ‘work type’ is included in the Work dimension. Requests, orders and order steps can be related through the Work Relationship table, which allows work order steps to be aggregated into one order, or work orders to be aggregated into work requests.

Pre-calculated time durations and differences are included in the fact table to simplify queries, and numeric and character data are included. While the attribute `Repeat Interval` appears to be a likely candidate for inclusion in the Work dimension, it is more of a candidate for data analysis in looking at patterns of repeat intervals over various fact records.

MULTIDIMENSIONAL MODEL QUALITY

In order to evaluate the quality of the multidimensional schema produced, metrics must be defined that address characteristics of the schema. There are several papers that address data warehouse quality, but few that address multidimensional model quality.

Calero et al. [11] approached measuring data warehouse quality through a set of numeric metrics based schema characteristics. However, the metrics that had a positive correlation with complexity were indicators that were relative – hence a set of data warehouses in a similar area is required for comparison. Jarke et al. [12] briefly addressed data warehouse schema quality through five criteria: correctness, completeness, minimality, traceability, and interpretability. Moody [13] expanded the list, albeit for ER models, to eight factors: completeness, integrity, flexibility, understandability, correctness, simplicity, integration, and implementability. Apart from correctness and simplicity which can evaluated by CASE tools, and implementability through physical implementation, all other factors are ascertained through subjective peer reviews.

As the peer review characteristics largely depend on the deployment of a data warehouse into a business setting, the qualitative component of the quality framework will not be undertaken. Using CASE tools to develop the model provides inherent syntactical correctness within the model. Implementability was tested through developing examples OLAP queries.

TESTING

Testing the multidimensional schemas involved measuring the query conceptualisation complexity. Decision support systems employ queries to extract relevant data. These queries may be defined by the developer of the system at design, or the system may allow for ad-hoc queries at runtime. In either case, the developer or the user is required to formalise the query according to the query language syntax. Less complex queries lead to shortened design time in additional to requiring a reduced technical capability required by the designer.

Five different query types were tested against the multidimensional and entity relationship models. The queries were written in SQL and included loosely and tightly constrained joins, calculations, aggregations, and sorting. As Microsoft SQL Server will be used in future testing, SQL functions were based on Transact-SQL.

Query type	Format	Lines of code	Joins	Internal functions	Other
Loosely constrained joins	ER	15	3	5	Requires subquery
	Multidimensional	6	2	2	
Tightly constrained joins	ER	26	7	5	Requires subquery
	Multidimensional	10	4	2	
Calculations	ER	5	1	2	
	Multidimensional	1	0	0	
Aggregations	ER	14	3	3	
	Multidimensional	8	2	3	
Sorting	ER	6	1	1	
	Multidimensional	5	1	1	

Table 1 – Query type characteristics

As can be seen in Table 1, multidimensional schemas resulted in the same or improved results for all query types. The lines of code were always smaller due to less joins between tables, and the use of aggregated facts. The number of internal functions used was also reduced because of pre-computed and aggregated facts. With the first two query types that tested joins, subqueries were required with the ER case but were unneeded with multidimensional schemas. While these are preliminary results, the above table shows improved query conceptualisation complexity when using multidimensional schemas.

CONCLUSIONS

Asset management data warehousing is an open area of research. Development of new technologies in asset management often leads to an increase in amount of data collected, therefore methods of arranging that data are always in need. Data warehousing is one possible methodology which is slowly starting to be examined by the engineering asset management community.

In this paper, we developed a methodology of turning the MIMOSA OSA-EAI CRIS into a multidimensional model for data warehousing. It was shown that in multidimensional form, it always lead to simplified query conceptualisation.

This research is an initial entry into a field that is very open to further work. Whether our results will translate into performance gains remains to be seen and future testing will be conducted on a case study.

REFERENCES

1. British Standards, *PAS 55-1: Asset management. Specification for the optimised management of physical infrastructure assets*. 2003: London, United Kingdom.
2. Mathew, A., et al. *A water utility industry conceptual asset management data warehouse model*. in *Computers and Industrial Engineering*. 2006. Taipei, Taiwan.
3. Boehnlein, M. and A.U. vom Ende. *Deriving initial data warehouse structures from the conceptual data models of the underlying operational information systems*. in *ACM International Workshop on Data Warehousing and OLAP*. 1999. Kansas City, Missouri, USA.
4. Bonifati, A., et al., *Designing data marts for data warehouses.* ACM Transactions on Software Engineering Methodology, 2001. **10**(4): p. 452-483.
5. Cabibbo, L. and R. Torlone. *A logical approach to multidimensional databases*. in *International Conference on Extending Database Technology: Advances in Database Technology*. 1998. Valencia, Spain: Springer.
6. Golfarelli, M., D. Maio, and S. Rizzi. *Conceptual design of data warehouses from E/R schemes*. in *Hawaii International Conference On System Sciences*. 1998. Kohala Coast, Hawaii, USA.
7. Moody, D.L. and M.A.R. Kortink. *From enterprise models to dimensional models: A methodology for data warehouse and data mart design*. in *Design and Management of Data Warehouses*. 2000. Stockholm, Sweden: CEUR-WS.org.
8. MIMOSA, *OSA-EAI Terminology Dictionary V3.0*. 2006.
9. Nadkarni, P.M., et al., *Organization of heterogeneous scientific data using the EAV/CR representation.* Journal of the American Medical Informatics Association, 1999. **6**(6): p. 478-493.
10. Mathew, A., et al. *A review of the MIMOSA OSA-EAI database for condition monitoring systems*. in *World Congress on Engineering Asset Management*. 2006. Gold Coast, Queensland, Australia: Springer.
11. Calero, C., et al. *Towards data warehouse quality metrics*. in *International Workshop on Design and Management of Data Warehouses*. 2001. Interlaken, Switzerland: CEUR-WS.org.
12. Jarke, M., et al., *Architecture and quality in data warehouses: An extended repository approach.* Information Systems, 1999. **24**(3): p. 229-253.
13. Moody, D.L. *Metrics for evaluating the quality of entity relationship models*. in *International Conference on Conceptual Modeling*. 1998. Singapore: Springer.